Making Vision Transformers Truly Shift-Equivariant

Renan A. Rojas-Gomez[1] Teck-Yian Lim[1] Minh N. Do[1,2] Raymond A. Yeh[3]
[1]Department of Electrical Engineering, UIUC [2]VinUni-Illinois Smart Health Center, UIUC
[3]Department of Computer Science, Purdue University
{renanar2, tlim11, minhdo}@illinois.edu rayyeh@purdue.edu

Abstract

In the field of computer vision, Vision Transformers (ViTs) have emerged as a prominent deep learning architecture. Despite being inspired by Convolutional Neural Networks (CNNs), ViTs are susceptible to small spatial shifts in the input data – they lack shift-equivariance. To address this shortcoming, we introduce novel data-adaptive designs for each of the ViT modules that break shift-equivariance, such as tokenization, self-attention, patch merging, and positional encoding. With our proposed modules, we achieve perfect circular shift-equivariance across four prominent ViT architectures: Swin, SwinV2, CvT, and MViTv2. Additionally, we leverage our design to further enhance consistency under standard shifts. We evaluate our adaptive ViT models on image classification and semantic segmentation tasks. Our models achieve competitive performance across three diverse datasets, showcasing perfect (100%*) circular shift consistency while improving standard shift consistency.*[1]

1. Introduction

Vision Transformers (ViTs) [11, 12, 20, 24, 25, 48] have become a strong alternative to convolutional neural networks (CNNs) in computer vision, superseding their dominance in image classification and becoming the state-of-the-art model on ImageNet [9]. Unlike the original Transformer [44] proposed for natural language processing (NLP), ViTs incorporate suitable inductive biases for computer vision. Consider image classification, where an input shift does not change the underlying image label, *i.e.*, the task is shift-invariant.

Several ViTs accredited shift-invariance as the motivation for the proposed architecture. Wu et al. [48] state that their ViT model brings "desirable properties of CNNs to the ViT architecture (i.e. *shift*, scale, and distortion invariance)." Similarly, Liu et al. [24] found that "inductive bias that encourages certain translation invariance is still preferable for general-purpose visual modeling." While existing ViTs incorporate such design elements, they still exhibit sensitivity to spatial input shifts. This motivates us to explore design principles towards shift-invariance and equivariance in ViTs.

This work delves into the core building blocks of ViTs and introduces a novel framework that guarantees perfect circular shift-equivariance within each module. This encompasses redesigned versions of the tokenization, self-attention, patch merging, and positional encoding modules. Our approach involves performing an input-dependent alignment, meaning each module's behavior adapts to the input. Consequently, we denote our modules as *(A)daptive*. We rigorously show that our adaptive modules are provably circularly shift-equivariant and realizable in practical scenarios.

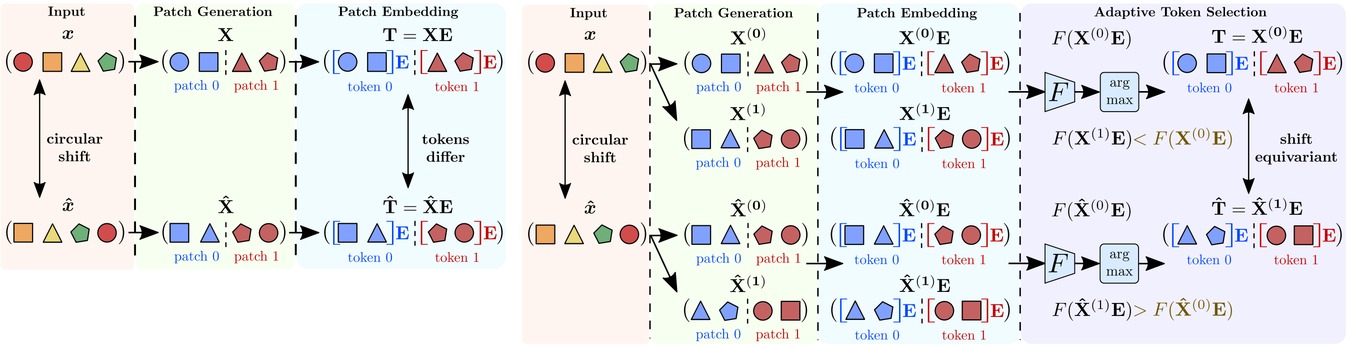
The proposed data-adaptive design enables the construction of truly circularly shift-invariant ViTs for image classification and truly circularly shift-equivariant ViTs for semantic segmentation, achieving 100% circular shift consistency. Furthermore, it fosters improvements in standard shift consistency while maintaining competitive performance on both image classification and semantic segmentation tasks.

To demonstrate the practical value of our framework, we conduct experiments across various ViT architectures and datasets. These include well-established benchmarks for image classification (CIFAR-10/100 [18] and ImageNet) and semantic segmentation (ADE20K [56]). We empirically show that our design improves shift consistency and achieve competitive performance on four prominent ViT architectures: Swin [24], SwinV2 [25], CvT [48], and MViTv2 [20].

Our contributions are as follows:

- We introduce a data-adaptive design for key ViT modules – tokenization, self-attention, patch merging, and positional encoding – provably achieving circular shift-equivariance.
- By leveraging our adaptive modules, we construct ViT models that achieve perfect (100%) end-to-end circular shift consistency, while also improving standard shift consistency, as shown on four established architectures.
- Extensive image classification and semantic segmentation experiments showcase the effectiveness of our data-adaptive design in achieving improved consistency and accuracy under both circular and standard shifts.

[1] Project website: https://renanrojasg.github.io/shifteq_vit.


Input
$\boldsymbol{x}$
Patch Generation
$\mathbf{X}$
patch 0
patch 1
Patch Embedding
$\mathbf{T} = \mathbf{XE}$
token 0
token 1
circular shift
tokens differ
$\hat{\boldsymbol{x}}$
$\hat{\mathbf{X}}$
$\hat{\mathbf{T}} = \hat{\mathbf{X}}\mathbf{E}$
$\mathbf{X}^{(0)}$
$\mathbf{X}^{(1)}$
$\mathbf{X}^{(0)}\mathbf{E}$
$\mathbf{X}^{(1)}\mathbf{E}$
Adaptive Token Selection
$F(\mathbf{X}^{(0)}\mathbf{E})$
$\mathbf{T} = \mathbf{X}^{(0)}\mathbf{E}$
F
arg max
$F(\mathbf{X}^{(1)}\mathbf{E}) < F(\mathbf{X}^{(0)}\mathbf{E})$
shift equivariant
$\hat{\mathbf{X}}^{(0)}$
$\hat{\mathbf{X}}^{(1)}$
$\hat{\mathbf{X}}^{(0)}\mathbf{E}$
$\hat{\mathbf{X}}^{(1)}\mathbf{E}$
$F(\hat{\mathbf{X}}^{(0)}\mathbf{E})$
$\hat{\mathbf{T}} = \hat{\mathbf{X}}^{(1)}\mathbf{E}$
$F(\hat{\mathbf{X}}^{(1)}\mathbf{E}) > F(\hat{\mathbf{X}}^{(0)}\mathbf{E})$

(a) Original tokenization (`token`).

(b) Proposed adaptive tokenization (`A-token`).

Figure 1. **Re-designing ViT's tokenization towards shift-equivariance:** (a) The original patch embedding is sensitive to small input shifts due to the fixed grid used to split an image into patches. (b) Our adaptive tokenization `A-token` is a generalization that consistently selects the group of patches with the highest energy, despite circular input shifts.

2. Related Work

Vision transformers. Originally designed for NLP tasks, Transformers [44] combine tokenization, positional encoding, and attention mechanisms in a novel architecture. This was later adapted for computer vision by incorporating inductive biases like shift equivariance, giving rise to the area of Vision Transformers. Seminal works include: ViT [11], which splits images into 16 × 16 tokens; Swin [24, 25], which uses localized attention; CvT [48], which integrates convolutional layers; and MViT [12, 20], with its multi-scale pyramid structure. Our work re-examines ViTs' modules and presents a novel design that achieves perfect circular shift-equivariance and enhances standard shift-equivariance.

Recent work proposes an *anchoring* method [10] to achieve circularly shift-invariant ViTs for classification. Similar to CNN techniques [3], it relies on the polyphase decomposition to align images before feeding them to a ViT. In contrast, we redesign all modules that break shift-equivariance, obtaining an end-to-end circularly shift-equivariant ViT. Our design achieves improved circular and linear shift consistency in both classification and semantic segmentation tasks.

Equivariant and invariant CNNs. Prior work [1, 55] have shown that modern CNNs [14, 19, 39, 42] are not shift-equivariant due to the usage of pooling layers. To improve shift-equivariance, Zhang [55] and Zou et al. [57] propose using lowpass filters (LPF) for anti-aliasing purposes [46].

While anti-aliasing improves shift consistency, the overall CNN remains not shift-equivariant. To address this, Chaman and Dokmanic [3] propose Adaptive Polyphase Sampling (APS), which leverages the input's polyphase decomposition to achieve circular shift-equivariance. Rojas-Gomez et al. [34] improve on APS by proposing a Learnable Polyphase Sampling (LPS) layer that imposes circular shift-equivariance. In contrast, our adaptive modules improve shift-equivariance in ViTs. Notice that CNN methods are *not applicable* to ViTs due to their distinct architectures.

Beyond shift-equivariance, broader research studies general equivariance [2, 4, 17, 28, 31, 32, 36, 37, 40, 43, 45, 47, 52]. Equivariant networks extend to sets [13, 27, 30, 33, 50, 54], graphs [7, 8, 16, 22, 23, 26, 29, 41, 51], among others.

3. Preliminaries

We review the basics before introducing our approach, including the aspects of current ViTs that break shift-equivariance. For readability, the concepts are described in 1D. In practice, these are extended to multi-channel images.

Equivariance. Conceptually, equivariance describes a function's input and output relation under *predefined transformations*. For example, in image segmentation, shift equivariance means that *shifting* the input results in *shifting* the output mask. Following previous work [3, 34], our analysis focuses on shift equivariance under *circular shifts*, denoted as:

$$(\mathcal{S}_N \boldsymbol{x})[n] = \boldsymbol{x}[(n+1) \bmod N], \boldsymbol{x} \in \mathbb{R}^N. \tag{1}$$

This ensures that a shifted signal $\boldsymbol{x}$ remains within its support. Following Rojas-Gomez et al. [34], we say a function $f : \mathbb{R}^N \mapsto \mathbb{R}^M$ is $\mathcal{S}_N, \{\mathcal{S}_M, \boldsymbol{I}\}$-equivariant or shift-equivariant iff $\exists\, \mathcal{S} \in \{\mathcal{S}_M, \boldsymbol{I}\}$ s.t.

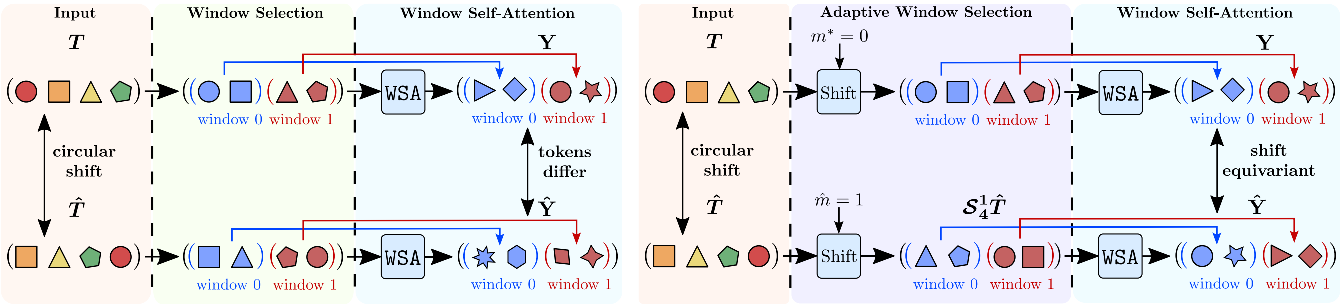
$$f(\mathcal{S}_N \boldsymbol{x}) = \mathcal{S} f(\boldsymbol{x}) \; \forall \boldsymbol{x} \in \mathbb{R}^N, \tag{2}$$

where $\boldsymbol{I}$ denotes the identity mapping. This definition carefully handles the case where $N > M$. For instance, when downsampling by a factor of two, an input shift by one should ideally induce an output shift by 0.5, which is not realizable on the integer grid. This 0.5 has to be rounded up or down, hence a shift $\mathcal{S}_M$ or a no-shift $\boldsymbol{I}$, respectively.

Invariance. For classification, a label remains *unchanged* when the image is *shifted*, *i.e.*, it is shift-invariant. A function $f : \mathbb{R}^N \mapsto \mathbb{R}^M$ is $\mathcal{S}_N, \{\boldsymbol{I}\}$-equivariant or shift-invariant iff:

$$f(\mathcal{S}_N \boldsymbol{x}) = f(\boldsymbol{x}) \; \forall \boldsymbol{x} \in \mathbb{R}^N. \tag{3}$$

A common way to design a shift-invariant function under circular shifts is via *global spatial pooling* [21], defined as



(a) Window-based self-attention (`WSA`)

(b) Proposed adaptive window-based self-attention (`A-WSA`)

Figure 2. **Re-designing window-based self-attention towards shift-equivariance:** (a) The window-based self-attention `WSA` breaks shift equivariance by selecting windows without considering their input properties. (b) Our proposed adaptive window-based self-attention selects the best grid of windows based on their average energy, obtaining windows comprised of the same tokens despite circular input shifts.

$g(\boldsymbol{x}) = \sum_m \boldsymbol{x}[m]$. Given a shift-equivariant function f:

$$\sum_m f(\mathcal{S}_N \boldsymbol{x})[m] = \sum_m \mathcal{S} f(\boldsymbol{x})[m] = \sum_m f(\boldsymbol{x})[m]. \qquad (4)$$

However, ViTs using global spatial pooling after extracting features are *not shift-invariant*, as layers such as tokenization, self-attention, and patch merging break shift-equivariance. **Tokenization (`token`).** ViTs split an input $\boldsymbol{x} \in \mathbb{R}^N$ into non-overlapping patches of length L and project them into a latent space to generate tokens. This operation is defined as:

$$\texttt{token}(\boldsymbol{x}) = \boldsymbol{X}\boldsymbol{E} \in \mathbb{R}^{\frac{N}{L} \times D}, \qquad (5)$$

where $\boldsymbol{X} = \texttt{reshape}(\boldsymbol{x}) = \left[\boldsymbol{X}_0 \ \dots \ \boldsymbol{X}_{\frac{N}{L}-1}\right]^\top \in \mathbb{R}^{\frac{N}{L} \times L}$ is comprised by non-overlapping patches of $\boldsymbol{x}$:

$$\boldsymbol{X}_k = \boldsymbol{x}[Lk : L(k+1) - 1] \in \mathbb{R}^L, \qquad (6)$$

and $\boldsymbol{E} \in \mathbb{R}^{L \times D}$. Note that `token` lacks shift-equivariance, since patches are extracted based on a *fixed* grid. So, different patches are obtained from shifted inputs, as shown in Fig. 1a. **Self-Attention (`SA`).** In ViTs, self-attention is defined as:

$$\texttt{SA}(\boldsymbol{T}) = \texttt{softmax}(\boldsymbol{Q}\boldsymbol{K}^\top / \sqrt{D'})\boldsymbol{V} \in \mathbb{R}^{M \times D'}, \qquad (7)$$

where $\boldsymbol{T} = \left[\boldsymbol{T}_0 \ \dots \ \boldsymbol{T}_{M-1}\right]^\top \in \mathbb{R}^{M \times D}$ denotes input tokens, and `softmax` is the softmax normalization along rows. Queries $\boldsymbol{Q}$, keys $\boldsymbol{K}$ and values $\boldsymbol{V}$ correspond to:

$$\boldsymbol{Q} = \boldsymbol{T}\boldsymbol{E}^Q, \ \boldsymbol{K} = \boldsymbol{T}\boldsymbol{E}^K, \ \boldsymbol{V} = \boldsymbol{T}\boldsymbol{E}^V, \qquad (8)$$

with linear projections $\boldsymbol{E}^{Q/K/V} \in \mathbb{R}^{D \times D'}$. The term $\texttt{softmax}\left(\boldsymbol{Q}\boldsymbol{K}^\top / \sqrt{D'}\right) \in [0,1]^{M \times M}$ ensures that the output token is a convex combination of the computed values. **Window-based self-attention (`WSA`).** A crucial limitation of self-attention is its quadratic computational cost with respect to the number of input tokens M. To alleviate this, window-based self-attention [24] groups tokens into local windows and then performs self-attention *within* each window. Given input tokens $\boldsymbol{T} \in \mathbb{R}^{M \times D}$ and a window size W, window-based self-attention $\texttt{WSA}(\boldsymbol{T}) \in \mathbb{R}^{M \times D'}$ is defined as:

$$\texttt{WSA}(\boldsymbol{T}) = \left[\texttt{SA}(\bar{\boldsymbol{T}}_W^{(0)}) \ ; \ \dots \ ; \ \texttt{SA}(\bar{\boldsymbol{T}}_W^{(\frac{M}{W}-1)})\right], \qquad (9)$$

where $\bar{\boldsymbol{T}}_W^{(k)} = \left[\boldsymbol{T}_{Wk} \ \dots \ \boldsymbol{T}_{W(k+1)-1}\right]^\top \in \mathbb{R}^{W \times D}$ is the k^{th} window comprising nearby tokens (W consecutive rows of $\boldsymbol{T}$). Notice that Eq. (9) use semicolons (;) as row separators.

Swin architectures [24, 25] take advantage of `WSA` to decrease the computational cost while adopting a shifting scheme (at the window level) to allow long-range connections. We note that `WSA` is not shift-equivariant, *e.g.*, any circular shift that is not a multiple of the window size changes the tokens within each window, as illustrated in Fig. 2a. **Patch merging (`PMerge`).** Given a patch length P and input tokens $\boldsymbol{T} = \left[\boldsymbol{T}_0 \ \dots \ \boldsymbol{T}_{M-1}\right]^\top \in \mathbb{R}^{M \times D}$, patch merging is defined as a linear projection of vectorized token patches:

$$\texttt{PMerge}(\boldsymbol{T}) = \tilde{\boldsymbol{T}}\tilde{\boldsymbol{E}} \in \mathbb{R}^{\frac{M}{P} \times \tilde{D}}, \qquad (10)$$

$$\text{with } \tilde{\boldsymbol{T}} = \left[\texttt{vec}(\bar{\boldsymbol{T}}_P^{(0)}) \ \dots \ \texttt{vec}(\bar{\boldsymbol{T}}_P^{(\frac{M}{P}-1)})\right]^\top.$$

Here, $\texttt{vec}(\bar{\boldsymbol{T}}_P^{(k)}) \in \mathbb{R}^{PD}$ is the vectorized version of the k^{th} patch $\bar{\boldsymbol{T}}_P^{(k)} = \left[\boldsymbol{T}_{Pk} \ \dots \ \boldsymbol{T}_{P(k+1)-1}\right]^\top \in \mathbb{R}^{P \times D}$, and $\tilde{\boldsymbol{E}} \in \mathbb{R}^{PD \times \tilde{D}}$ is a linear projection. `PMerge` reduces the number of tokens while increasing their length, *i.e.*, $\tilde{D} > D$. This follows the CNN strategy of increasing the number of channels using convolutional layers while decreasing the spatial resolution via pooling. Since patches are also selected using a fixed grid, `PMerge` breaks shift-equivariance. **Relative position embedding (RPE).** As self-attention is permutation equivariant, spatial information must be explicitly incorporated. Typically, RPE adds a position matrix representing the relative distance between queries and keys:

$$\texttt{SA}^{(\text{rel})}(\boldsymbol{T}) = \texttt{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^\top}{\sqrt{D'}} + \boldsymbol{E}^{(\text{rel})}\right)\boldsymbol{V}, \qquad (11)$$

$$\text{with } \boldsymbol{E}^{(\text{rel})}[i,j] = \boldsymbol{B}^{(\text{rel})}[p_i^{(Q)} - p_j^{(K)}]. \qquad (12)$$

Here, $\boldsymbol{E}^{(\text{rel})} \in \mathbb{R}^{M\times M}$ is built from an embedding lookup table $\boldsymbol{B}^{(\text{rel})} \in \mathbb{R}^{2M-1}$ and the index $[p_i^{(Q)} - p_j^{(K)}]$ denotes the distance between the i^{th} query token at position $p_i^{(Q)}$ and the j^{th} key token at position $p_j^{(K)}$. By relying on the relative token distance, RPE allows ViTs to capture spatial relationships, *e.g.*, knowing if two tokens are spatially nearby.

4. Circularly Shift-equivariant ViT

To achieve circular shift-equivariance, we redesign ViT's tokenization, self-attention, patch merging, and positional embedding modules. As equivariance is preserved under compositions, resulting ViTs become end-to-end shift-equivariant.

Adaptive tokenization (`A-token`). `Token` splits an input into patches in a fixed manner, breaking shift-equivariance. We propose a data-dependent alternative that selects patches that maximize a shift-invariant function, resulting in the same tokens regardless of input shifts. Given an input $\boldsymbol{x} \in \mathbb{R}^N$ and a patch length L, our adaptive tokenization is defined as:

$$\texttt{A-token}(\boldsymbol{x}) = \boldsymbol{X}^{(m^\star)}\boldsymbol{E} \in \mathbb{R}^{\frac{N}{L}\times D}, \tag{13}$$

$$\text{with } m^\star = \operatorname*{arg\,max}_{m\in\{0,\dots,L-1\}} F(\boldsymbol{X}^{(m)}\boldsymbol{E}). \tag{14}$$

Here, $\boldsymbol{X}^{(m)} = \texttt{reshape}(\mathcal{S}_N^m \boldsymbol{x}) \in \mathbb{R}^{\frac{N}{L}\times L}$ is the reshaped version of the input circularly shifted by m samples, $\boldsymbol{E} \in \mathbb{R}^{L\times D}$ is a linear projection and $F : \mathbb{R}^{\frac{N}{L}\times D} \mapsto \mathbb{R}$ is a shift-invariant function. Notice that $m \in \{0,\dots,L-1\}$ since the token representation of an input is only affected by circular shifts up to the patch size L. For any shift greater or equal than L, there is a shift smaller than L that generates the same tokens. So, an input can be represented in L distinct ways. Fig. 1b shows our circularly shift-equivariant tokenization.

`A-token` maximizes a shift-invariant function to ensure the same token representation regardless of circular input shifts. Next, we analyze a core property of $\boldsymbol{X}^{(m)}\boldsymbol{E}$ to prove that our adaptive tokenization is circularly shift-equivariant.

Lemma 1. L-periodic shift-equivariance of tokenization.
Let input $\boldsymbol{x} \in \mathbb{R}^N$ have a token representation $\boldsymbol{X}^{(m)}\boldsymbol{E} \in \mathbb{R}^{\lfloor N/L\rfloor\times D}$. If $\hat{\boldsymbol{x}} = \mathcal{S}_N\boldsymbol{x}$ (a shifted input), then its token representation $\hat{\boldsymbol{X}}^{(m)}\boldsymbol{E}$ corresponds to:

$$\hat{\boldsymbol{X}}^{(m)}\boldsymbol{E} = \mathcal{S}_{\lfloor N/L\rfloor}^{\lfloor (m+1)/L\rfloor}\boldsymbol{X}^{((m+1) \bmod L)}\boldsymbol{E}. \tag{15}$$

This implies that $\boldsymbol{x}$ and $\hat{\boldsymbol{x}}$ are characterized by the same L token representations, up to a circular shift along the token index (row index of $\boldsymbol{X}^{((m+1) \bmod L)}\boldsymbol{E}$).

Proof. By definition, $\hat{\boldsymbol{X}}^{(m)} = \texttt{reshape}(\mathcal{S}_N^{m+1}\boldsymbol{x})$. Expressing $m+1$ as quotient and remainder for divisor L, the remainder indicates matching token representations (representations of $\boldsymbol{x}$ and $\hat{\boldsymbol{x}}$ comprised of the same tokens), while the quotient indicates the one-to-one correspondence between their tokens. The proof is deferred to Appendix Sec. A1. □

Lemma 1 shows that, for any index $\hat{m}$, there exists $m = (\hat{m}+1) \bmod L$ such that $\hat{\boldsymbol{X}}^{(\hat{m})}$ and $\boldsymbol{X}^{(m)}$ are equal up to a circular shift. In Claim 1, we use this property to demonstrate the shift-equivariance of our proposed adaptive tokenization.

Claim 1. *Shift-equivariance of adaptive tokenization.*
If F in Eq. (14) is shift-invariant, then `A-token` is shift-equivariant, i.e., $\exists\, m_q \in \{0,\dots,L-1\}$ s.t.

$$\texttt{A-token}(\mathcal{S}_N\boldsymbol{x}) = \mathcal{S}_{\lfloor N/L\rfloor}^{m_q}\texttt{A-token}(\boldsymbol{x}). \tag{16}$$

Proof. Given $m^\star$ in Eq. (14) and Lemma 1, $\exists\, \hat{m}$ such that $\hat{\boldsymbol{X}}^{(\hat{m})}\boldsymbol{E}$ is a circularly shifted version of $\boldsymbol{X}^{(m^\star)}\boldsymbol{E}$. Since $\boldsymbol{x}$ and $\mathcal{S}_N\boldsymbol{x}$ have the same L token representations and given a shift-invariant F, we show $\hat{m}$ maximizes $F(\hat{\boldsymbol{X}}^{(m)}\boldsymbol{E})$. So, $\texttt{A-token}(\mathcal{S}_N\boldsymbol{x})$ is a circularly shifted version of $\texttt{A-token}(\boldsymbol{x})$. See Appendix Sec. A1 for the full proof. □

Adaptive window-based self-attention (`A-WSA`). `WSA`'s window partitioning is shift-sensitive, as different windows are obtained when the input tokens are circularly shifted by a *non-multiple of the window size*. We propose an adaptive token shifting method to obtain a consistent window partition. By selecting the offset based on the *energy* of all possible window partitions, our method generates the same windows regardless of circular shifts in the input tokens.

Given tokens $\boldsymbol{T} = \begin{bmatrix}\boldsymbol{T}_0 & \dots & \boldsymbol{T}_{M-1}\end{bmatrix}^\top \in \mathbb{R}^{M\times D}$ and a window size W, let $\boldsymbol{v}_W \in \mathbb{R}^{\lfloor\frac{M}{W}\rfloor}$ consist of the average ℓ_p-norm or energy of each window (W adjacent tokens):

$$\boldsymbol{v}_W[k] = \frac{1}{W}\sum_{l=0}^{W-1}\|\boldsymbol{T}_{(Wk+l) \bmod M}\|_p. \tag{17}$$

Then, the energy of the windows resulting from circularly shifting the input tokens by m indices corresponds to $\boldsymbol{v}_W^{(m)} \in \mathbb{R}^{\lfloor\frac{M}{W}\rfloor}$, where $\boldsymbol{v}_W^{(m)}[k]$ is the energy of the k^{th} window:

$$\boldsymbol{v}_W^{(m)}[k] = \frac{1}{W}\sum_{l=0}^{W-1}\|\underbrace{(\mathcal{S}_M^m\boldsymbol{T})_{(Wk+l) \bmod M}}_{=\boldsymbol{T}_{(Wk+m+l) \bmod M}}\|_p. \tag{18}$$

Based on the window energy in Eq. (18), we define the adaptive window-based self-attention as:

$$\texttt{A-WSA}(\boldsymbol{T}) = \texttt{WSA}\big(\mathcal{S}_M^{m^\star}\boldsymbol{T}\big) \in \mathbb{R}^{M\times D'}, \tag{19}$$

$$\text{with } m^\star = \operatorname*{arg\,max}_{m\in\{0,\dots,W-1\}} G\big(\boldsymbol{v}_W^{(m)}\big), \tag{20}$$

where $G : \mathbb{R}^{\lfloor\frac{M}{W}\rfloor} \mapsto \mathbb{R}$ is a shift-invariant function. By choosing windows based on $m^\star$, `A-WSA` generates the same group of windows despite input shifts, as shown in Claim 2.

Claim 2. *If G in Eq. (19) is shift invariant, then `A-WSA` is shift-equivariant.*

Proof. Given two groups of tokens related by a circular shift, and a shift-invariant function G, shifting each group by its maximizer in Eq. (19) induces an offset that is a multiple of W. So, both groups are partitioned in the same windows up to a circular shift. Fig. 2b illustrates this consistent window grouping. The proof is deferred to Appendix Sec. A1. □

Adaptive patch merging (`A-PMerge`). As shown in Sec. 3, `PMerge` consists of a vectorization of P neighboring tokens followed by a projection from $\mathbb{R}^{PD}$ to $\mathbb{R}^{\tilde{D}}$. So, it can be expressed as a strided convolution with $\tilde{D}$ output channels, stride factor P and kernel size P. We use this property to propose a circularly shift-equivariant patch merging.

Claim 3. *`PMerge` corresponds to a strided convolution with $\tilde{D}$ output channels, striding P and kernel size P.*

Proof. Expressing the linear projection $\tilde{\boldsymbol{E}}$ as a convolutional matrix, `PMerge` is equivalent to a convolution sum with kernels comprised by columns of $\tilde{\boldsymbol{E}}$. Let the input tokens be expressed as $\boldsymbol{T} = \begin{bmatrix}\boldsymbol{t}_0 & \dots & \boldsymbol{t}_{D-1}\end{bmatrix} \in \mathbb{R}^{M \times D}$, where $\boldsymbol{t}_j \in \mathbb{R}^M$ corresponds to the j^{th} element of every input token. Then, $\texttt{PMerge}(\boldsymbol{T}) \in \mathbb{R}^{\frac{M}{P} \times \tilde{D}}$ can be expressed as:

$$\texttt{PMerge}(\boldsymbol{T}) = \mathcal{D}^{(P)}(\begin{bmatrix}\boldsymbol{y}_0 & \dots & \boldsymbol{y}_{\tilde{D}-1}\end{bmatrix}), \quad (21)$$

$$\text{with } \boldsymbol{y}_k = \sum_{j=0}^{D-1} \boldsymbol{t}_j \circledast \boldsymbol{h}^{(k,j)} \in \mathbb{R}^M, \quad (22)$$

where $\mathcal{D}^{(P)} \in \mathbb{R}^{\frac{M}{P} \times M}$ is a striding operator of factor P, $\circledast$ denotes circular convolution and $\{\boldsymbol{h}^{(k,j)}\}_{k,j}$ are kernels of length P. Proof is deferred to Appendix Sec. A1. □

Following Claim 3, to attain circular shift-equivariance, we adopt APS [3] + LPF [55] as the striding operator. Let $\texttt{APS}^{(P)}$ denote the polyphase sampling layer of striding factor P. Then, $\texttt{A-PMerge}(\boldsymbol{T}) \in \mathbb{R}^{\frac{M}{P} \times \tilde{D}}$ corresponds to:

$$\texttt{A-PMerge}(\boldsymbol{T}) = \texttt{APS}^{(P)}(\begin{bmatrix}\boldsymbol{y}_0 & \dots & \boldsymbol{y}_{\tilde{D}-1}\end{bmatrix}). \quad (23)$$

Specifically, `APMerge` achieves circular shift-equivariance by adaptively choosing $\frac{M}{P}$ tokens based on their ℓ_2 norm.

Adaptive RPE. While the original relative distance matrix $\boldsymbol{E}^{(\text{rel})}$ is computed by taking into account linear shifts, this does not match our circular shift assumption; See Fig. 3 for a visualization. To obtain circular shift-equivariance, relative distances must consider the periodicity induced by circular shifts. Hence, we propose the adaptive relative position matrix $\boldsymbol{E}^{(\text{adapt})} \in \mathbb{R}^{M \times M}$, where each entry is defined as:

$$\boldsymbol{E}^{(\text{adapt})}[i,j] = \boldsymbol{B}^{(\text{adapt})}\left[(p_i^{(Q)} - p_j^{(K)}) \bmod M\right], \quad (24)$$

to encode the distance between the i^{th} query token at position $p_i^{(Q)}$ and the j^{th} key token at position $p_j^{(K)}$. Here, $\boldsymbol{B}^{(\text{adapt})} \in \mathbb{R}^M$ is the trainable lookup table comprised by relative positional embeddings. Notice that $\boldsymbol{B}^{(\text{adapt})}$ is smaller than the original $\boldsymbol{B}^{(\text{rel})} \in \mathbb{R}^{(2M-1)}$, since relative distances are now measured in a circular fashion between M tokens.

Segmentation with equivariant upsampling. Segmentation models with ViT backbones still rely on CNN decoders, *e.g.*, Swin [24] uses UperNet [49] as segmentation head. As shown in previous work [34], achieving circular shift-equivariance in CNN decoders requires keeping track of the pooling indices to put features back to their original positions during upsampling. Unlike CNNs, our ViT models also use an adaptive window selection. So, keeping track of window indices and accounting for their shifts becomes crucial to obtain circularly shift-equivariant ViT-based segmenters.

5. Experiments

We conduct experiments on image classification and semantic segmentation on four ViT architectures: Swin [24], SwinV2 [25], CvT [48], and MViTv2 [20]. We evaluate their performance under circular and standard shifts. For circular shifts, the experiments match our theory, so our models achieve 100% circular shift consistency (up to numerical errors). We further run experiments on standard shifts to study our method's performance under this theory-to-practice gap, where there is loss of information at the image boundaries.

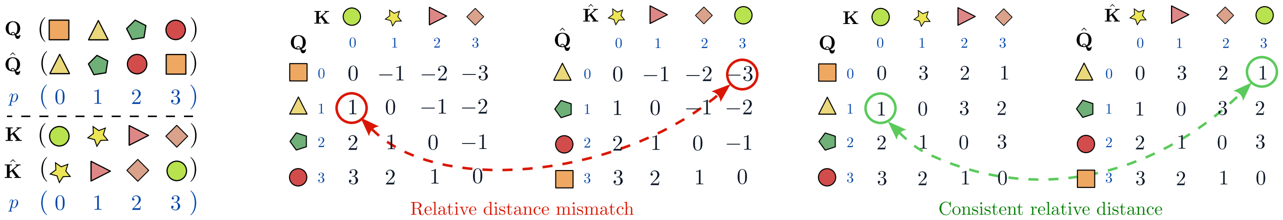
As detailed in Section 4, `A-token` and `A-WSA` use shift-invariant functions F and G, respectively, to ensure consistent token representations under circular shifts. After extensive benchmarking, in both cases, we selected the ℓ_p-norm as shift-invariant function. This allows a fast energy computation and diminishes problems caused by seemingly identical energy values due to numerical precision limitations.

5.1. Image classification under circular shifts

Experiment setup. We conduct experiments on CIFAR-10/100 [18], and ImageNet [9]. In all cases, images are resized to the resolution used by each model's original implementation (224 × 224 for Swin-T, CVT-13, and MViTv2-T; 256 × 256 for SwinV2-T). This allows us to use the same architecture across all datasets, *i.e.*, everything follows the original number of layers and blocks. To avoid boundary conditions, circular padding is used in all convolutional layers, and circular shifts are used for evaluating shift consistency.

On CIFAR-10/100, all models were trained for 100 epochs on two GPUs with batch size 48. The scheduler settings of each model were scaled accordingly. On ImageNet, all models were trained for 300 epochs on eight GPUs using their default batch sizes. Refer to Sec. A3 for full experimental details. For CIFAR-10/100, we report average and standard deviation metrics over five seeds. Due to computational limitations, we report on a single seed for ImageNet.

Evaluation metric. We report the top-1 classification accuracy on the original dataset without any shifts. To quantify shift-invariance, we report the circular shift consistency (C-Cons.), which counts how often the predicted labels are



(a) Shifted tokens (b) Original relative distance (c) Proposed relative distance

Figure 3. **Shift consistent relative distance:** (a) Circularly shifted queries and keys $(M = 4)$. (b) Original relative distance used to build the RPE matrix: $p^{(Q)}[i] - p^{(K)}[j]$. Since it does not consider the periodicity of circular shifts, relative distances are not preserved. (c) Proposed relative distance: $(p^{(Q)}[i] - p^{(K)}[j]) \bmod M$. Our proposed distance is consistent with circular shifts, leading to a shift equivariant RPE.

Method	Circular Shift				Standard Shift			
	CIFAR10		CIFAR-100		CIFAR10		CIFAR-100	
	Top-1 Acc.	C-Cons.	Top-1 Acc.	C-Cons.	Top-1 Acc.	S-Cons.	Top-1 Acc.	S-Cons.
Swin-T	90.15 ± .18	83.30 ± .61	71.01 ± .27	65.32 ± .69	90.11 ± .21	86.35 ± .25	71.12 ± .14	69.39 ± .52
A-Swin-T **(Ours)**	**93.39 ± .12**	**99.99 ± .01**	**75.11 ± .10**	**99.99 ± .01**	**93.50 ± .19**	**96.00 ± .08**	**75.12 ± .28**	**87.70 ± .57**
SwinV2-T	89.08 ± .21	89.16 ± .08	69.78 ± .22	75.23 ± .20	89.08 ± .21	91.68 ± .25	69.67 ± .32	80.42 ± .41
A-SwinV2-T **(Ours)**	**91.64 ± .21**	**99.99 ± .01**	**72.73 ± .23**	**99.96 ± .01**	**91.91 ± .12**	**95.81 ± .17**	**72.98 ± .13**	**88.74 ± .40**
CvT-13	90.06 ± .23	75.80 ± 1.2	66.61 ± .33	50.29 ± 1.68	90.05 ± .20	84.66 ± 1.26	66.06 ± .39	63.03 ± .73
A-CvT-13 **(Ours)**	**93.87 ± .14**	**100 ± .00**	**76.19 ± .32**	**100 ± .00**	**93.71 ± .10**	**96.47 ± .21**	**73.04 ± .23**	**86.96. ± .55**
MViTv2-T	96.00 ± .06	86.55 ± 1.2	80.18 ± .34	74.82 ± .73	96.14 ± .06	91.34. ± 1.26	80.28 ± .38	77.92. ± .93
A-MViTv2-T **(Ours)**	**96.41 ± .22**	**100 ± .00**	**81.39 ± .11**	**100 ± .00**	**96.61 ± .11**	**98.36. ± .16**	**81.17 ± .18**	**92.95. ± .16**

Table 1. **CIFAR-10/100 classification results:** Top-1 accuracy and shift consistency (%) under circular and standard shifts. Bold numbers indicate improvement over the corresponding baseline architecture. Mean and standard deviation reported over five random seeds.

identical under two different circular shifts. Given a dataset $\mathcal{D} = \{\boldsymbol{I}\}$, C-Cons. computes:

$$\frac{1}{|\mathcal{D}|} \sum_{\boldsymbol{I} \in \mathcal{D}} \mathbb{E}_{\boldsymbol{\Delta}_1, \boldsymbol{\Delta}_2} \left[\mathbf{1} \left[\hat{y}(\mathcal{S}^{\boldsymbol{\Delta}_1}(\boldsymbol{I})) = \hat{y}(\mathcal{S}^{\boldsymbol{\Delta}_2}(\boldsymbol{I})) \right] \right], \quad (25)$$

where $\mathbf{1}$ denotes the indicator function, $\hat{y}(\boldsymbol{I})$ the class prediction for $\boldsymbol{I}$, $\mathcal{S}$ the circular shift operator, and $\boldsymbol{\Delta}_1 = (h_1, w_1), \boldsymbol{\Delta}_2 = (h_2, w_2)$ horizontal and vertical offsets.

Results. We report performance in Tab. 1 and Tab. 2 for CIFAR-10/100 and ImageNet, respectively. Overall, we observe that our adaptive ViTs achieve near 100% shift consistency in practice. The remaining inconsistency is caused by errors inherent in numerical precision and tie-breaking that can lead to a wrong selection of tokens or windows. Beyond consistency improvements, our method also improves classification accuracy across all scenarios.

5.2. Image classification under standard shifts

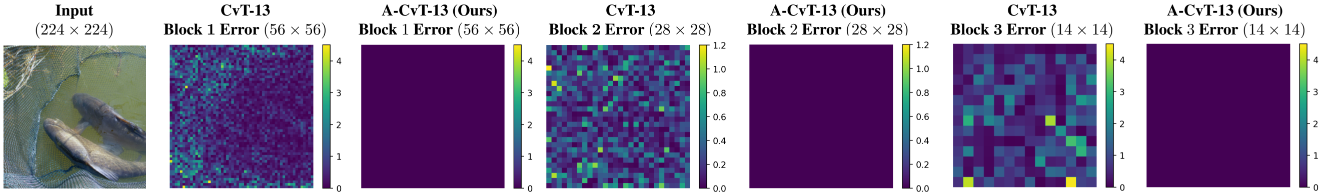
Experiment setup. To study the boundary effect on shift-invariance, we further conduct experiments using standard shifts. As these are no longer circular, the image content may change at its borders, *i.e.*, perfect shift consistency is *no longer guaranteed*. For CIFAR-10/100, input images were resized to the resolution used by each model's original implementation. Default data augmentation and optimizer settings were used for each model while training epochs and batch size followed those used in the circular shift settings.

Evaluation metric. We report top-1 classification accuracy on the original dataset (without any shifts). To quantify shift-invariance, we report the standard shift consistency (S-Cons.), which follows the same principle as C-Cons in Eq. (25), but uses a standard shift instead of a circular one. For CIFAR-10/100, we use zero-padding at the boundaries due to the small image size. For ImageNet, following Zhang [55], we perform an image shift followed by a center-cropping of size 224 × 224. This produces realistic shifts and avoids a particular choice of padding.

Method	Circular Shift		Standard Shift	
	Top-1 Acc.	C-Cons.	Top-1 Acc.	S-Cons.
Swin-T	78.5	86.68	81.18	92.41
A-Swin-T **(Ours)**	**79.35**	**99.98**	**81.6**	**93.24**
SwinV2-T	78.95	87.68	81.76	93.24
A-SwinV2-T **(Ours)**	**79.91**	**99.98**	**82.10**	**94.04**
CvT-13	77.01	86.87	**81.59**	92.80
A-CvT-13 **(Ours)**	**77.05**	**100**	81.48	**93.41**
MViTv2-T	77.36	90.03	82.21	93.88
A-MViTv2-T **(Ours)**	**77.46**	**100**	**82.4**	**94.08**

Table 2. **ImageNet classification results:** Top-1 accuracy and shift consistency (%) under circular and standard shifts. Bold numbers indicate improvement over the corresponding baseline architecture.

Results. Tabs. 1 and 2 report performance under standard shifts on CIFAR-10/100 and ImageNet, respectively. Due to boundary conditions, our method does not achieve 100% shift consistency. However, our adaptive models consistently outperform their baselines in terms of S-Cons. Our models also achieve higher classification performance in all settings except for CvT on ImageNet. Results highlight the practical value of our data-adaptive approach despite the gap in theory.



Input (224 × 224)
CvT-13
Block 1 Error (56 × 56)
A-CvT-13 (Ours)
Block 1 Error (56 × 56)
CvT-13
Block 2 Error (28 × 28)
A-CvT-13 (Ours)
Block 2 Error (28 × 28)
CvT-13
Block 3 Error (14 × 14)
A-CvT-13 (Ours)
Block 3 Error (14 × 14)

Figure 4. **Consistent token representations.** Shifting inputs by a small offset leads to large deviations (non-zero errors) in the representations when using default ViTs (*e.g.*, CvT-13). In contrast, our proposed models (*e.g.*, A-CvT-13) achieve an absolute zero-error across all blocks.

Model	# Params	Throughput (images/s)	Relative change (%)
Swin-T	28M	704.07	–
A-Swin-T **(Ours)**	28M	633.35	10.04
SwinV2-T	28M	470.81	–
A-SwinV2-T **(Ours)**	28M	405.01	13.98
CvT-13	20M	535.5	–
A-CvT-13 **(Ours)**	20M	492.12	10.69
MViTv2-T	24M	439.5	–
A-MViTv2-T **(Ours)**	24M	352.06	19.9

Table 3. **Inference throughput:** Absolute inference throughput (images/s) of our adaptive ViTs and their default versions. *Relative change* shows the throughput decrease w.r.t. the default models.

Module	Abs. runtime (ms)	Delta (ms)
Tokenization	8.37	–
A. Tokenization **(Ours)**	35.89	+27.52
Patch Merging {S2, S3, S4}	0.47, 0.45, 0.45	–
A. Patch Merging **(Ours)**	7.68, 4.47, 3.09	+7.21, +4.02, +2.64
Window Selection	Not applied	–
A. Window Selection **(Ours)**	7.63	+7.63
RPE	2.84	–
A. RPE **(Ours)**	9.91	+7.07

Table 4. **Runtime of adaptive ViT modules:** Inference runtime of our adaptive ViT modules and their default versions. *Delta* indicates the absolute time difference w.r.t. the default modules.

5.3. Consistency of tokens to input shifts

We evaluate the effect of small circular input shifts in the tokens obtained by our adaptive models. We verify the stability of our A-CvT-13 model by applying a circular shift of 1 row and 1 column to the input image, computing its tokens, and calculating their absolute difference to those of the unshifted image. Fig. 4 shows the absolute token difference of an ImageNet test sample at all three blocks of A-CvT-13, each with a different resolution. Similar to previous work [3], we illustrate errors for the channels with the highest energy.

In contrast to the large deviations of the default CvT-13 caused by the input shift, the tokens generated by our proposed A-CvT-13 model remain unaltered, as theoretically shown, leading to a circularly shift-equivariant ViT model.

5.4. Throughput and runtime analysis

We evaluate the inference throughput, measured in processed images per second, of our adaptive ViTs and modules over 100 forward passes (batch size 128, default image size per model) on a single NVIDIA Quadro RTX 5000 GPU.

Backbone	Circular Shift		Standard Shift	
	mIoU	mASCC	mIoU	mASSC
Swin-T	42.93	87.32	44.2	93.37
A-Swin-T **(Ours)**	**43.44**	**100**	**44.43**	**93.48**
SwinV2-T	43.86	88.16	44.26	93.23
A-SwinV2-T **(Ours)**	**44.42**	**100**	**46.11**	**93.59**

Table 5. **Semantic segmentation performance:** Segmentation accuracy and shift consistency (%) of our adaptive UperNet model equipped with A-Swin and A-SwinV2 backbones.

Model inference. We report the throughput of our adaptive ViTs and their default versions. We also measure the relative change, which corresponds to the throughput decrease with respect to the default ViT models.

Tab. 3 shows our adaptive models exhibit less than a 20% decrease in throughput w.r.t. the default models, while improving in shift consistency and classification accuracy without increasing the number of trainable parameters.

Modules runtime. We compare the runtime of our adaptive modules to that of the default ones. Tokenization, patch merging and window selection are evaluated on A-Swin, while RPE is evaluated on A-MViTv2 (A-Swin windows are comprised of the same tokens, so its RPE remains unaltered).

Tab. 4 shows the runtime of our adaptive modules, which slightly increases over the default runtime. This is particularly true for the adaptive RPE, where the main difference lies in the distance interpretation (circular vs. linear). While our adaptive tokenization has the largest increase by operating on full-size images, subsequent patch merging modules operate on smaller representations and are more efficient.

5.5. Semantic segmentation under circular shifts

Experiment setup. We conduct semantic segmentation experiments on the ADE20K dataset [56] using A-Swin and A-SwinV2 models as backbones and compare them against their default versions. Following previous work [24], we use UperNet [49] as the segmentation decoder. Similar to our classification settings under circular shifts, all convolutional layers in the UperNet model use circular padding to avoid boundary conditions, and circular shifts are used to measure shift consistency. Models are trained for 160K iterations on a total batch size of 16 using the default augmentation.

Evaluation metric. For segmentation performance, we re-

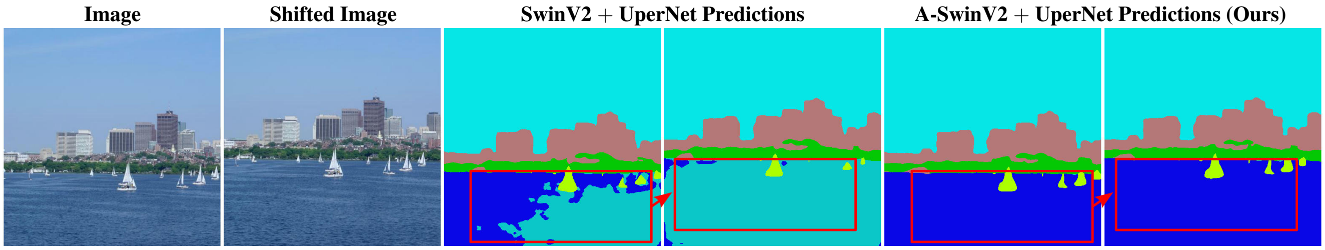


Image
Shifted Image
SwinV2 + UperNet Predictions
A-SwinV2 + UperNet Predictions (Ours)

Figure 5. **Segmentation under standard shifts:** Our A-SwinV2 + UperNet model improves robustness to input shifts over the original model, generating consistent predictions while improving accuracy. Examples of prediction changes due to shifts are highlighted in red.

port the mean intersection over union (mIoU) on the original dataset (without any shifts). For shift-equivariance, we report the mean-Average Segmentation Circular Consistency (mASCC) which counts how often the predicted pixel labels (after shifting back) are identical under two different circular shifts. Given a dataset $\mathcal{D} = \{\boldsymbol{I}\}$, mASCC computes

$$\frac{1}{|\mathcal{D}|}\sum_{\boldsymbol{I}\in\mathcal{D}}\mathbb{E}_{\boldsymbol{\Delta}_1,\boldsymbol{\Delta}_2}\left[\frac{1}{HW}\sum_{u=1,v=1}^{H,W}\mathbf{1}\left[\mathcal{S}^{-\boldsymbol{\Delta}_1}\hat{y}(\mathcal{S}^{\boldsymbol{\Delta}_1}(\boldsymbol{I}))[u,v]=\mathcal{S}^{-\boldsymbol{\Delta}_2}\hat{y}(\mathcal{S}^{\boldsymbol{\Delta}_2}(\boldsymbol{I}))[u,v]\right]\right], \tag{26}$$

where H, W correspond to the image height and width, and $[u, v]$ indexes the class prediction at pixel (u, v).

Results. Tab. 5 shows classification accuracy and shift consistency for UperNet segmenters using Swin-T and SwinV2-T backbones. Following the theory, our adaptive models achieve 100% mASCC (perfect circular shift consistency), while improving on segmentation accuracy.

5.6. Semantic segmentation under standard shifts

Experiment setup. As in the circular shift scenario, models are trained for 160K iterations with a total batch size of 16 using the default data augmentation. To evaluate shift-equivariance under standard shifts, we report the mean-Average Semantic Segmentation Consistency (mASSC), which counts how often the predicted pixel labels (after shifting back) are equal under two different standard shifts. Notice that mASSC ignores the boundary pixels in its computation as standard shifts lead to changes in boundary content.

Results. Tab. 5 shows results on the standard shift scenario. As anticipated, changes at image boundaries prevent perfect shift-equivariance. Regardless, our models improve segmentation accuracy and shift consistency, with a notable improvement on SwinV2-T. See Fig. 5 for segmentation results.

5.7. Ablation study

We study the impact of our adaptive ViT framework by systematically removing individual modules. Ablations are conducted on our A-Swin-T model trained on CIFAR-10 under circular shifts. Accuracy and circular shift consistency mean and standard deviation are computed over five seeds.

Configuration	Top-1 Acc.	C-Cons.
A-Swin-T **(Ours)**	$93.39 \pm .13$	**100**
(i) No `A-token`	$\mathbf{93.66} \pm \mathbf{.19}$	$96.29 \pm .20$
(ii) No `A-WSA`	$93.24 \pm .15$	$95.62 \pm .54$
(iii) No `A-PMerge`	$91.67 \pm .10$	$94.62 \pm .11$
Swin-T (Default)	$90.15 \pm .18$	$83.30 \pm .61$

Table 6. **Ablation study:** Effect of our shift-equivariant ViT modules on classification accuracy and shift consistency (%). Configurations progressively evaluated on Swin-T under circular shifts.

Results are reported in Tab. 6. Our full model improves circular shift consistency by more than 3.5% over A-Swin-T without `A-token`, while slightly decreasing classification accuracy by 0.27%. The use of `A-WSA` improves both classification accuracy and shift consistency. Finally, `A-PMerge` improves classification accuracy by approximately 1.7% and shift consistency by more than 5%. Overall, all adaptive modules are needed to achieve 100% circular shift consistency.

6. Conclusion

We propose a family of ViTs that are circularly shift-invariant and equivariant. We redesigned four ViT modules: tokenization, self-attention, patch merging, and relative position embedding to guarantee circular shift-invariance and equivariance theoretically. Leveraging these modules, we construct data-adaptive versions of prominent ViTs, making them end-to-end circularly shift-equivariant. When matching our theoretical setup, these models exhibit perfect (100%) circular shift consistency and outperform their baselines on image classification and segmentation. Furthermore, under standard shifts where image boundaries deviate from our assumptions, our adaptive models remain more resilient to input shifts. Notably, they maintain task performance on par with or exceeding the baselines, highlighting the practical value of our design.

Acknowledgments. This work was partially funded by the Jump ARCHES endowment through HCESC at UIUC. Our research used the Delta advanced computing, supported by the NSF award OAC 2005572 and the State of Illinois.

References

[1] A. Azulay and Y. Weiss. Why do deep convolutional networks generalize so poorly to small image transformations? *JMLR*, 2019. 2

[2] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst. Geometric deep learning: going beyond euclidean data. *IEEE SPM*, 2017. 2

[3] A. Chaman and I. Dokmanic. Truly shift-invariant convolutional neural networks. In *Proc. CVPR*, 2021. 2, 5, 7, 15, 19, 21

[4] T. Cohen and M. Welling. Group equivariant convolutional networks. In *Proc. ICML*, 2016. 2

[5] M. Contributors. MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark. https://github.com/open-mmlab/mmsegmentation, 2020. 21

[6] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proc. CVPR workshop*, 2020. 19

[7] P. de Haan, M. Weiler, T. Cohen, and M. Welling. Gauge equivariant mesh CNNs: Anisotropic convolutions on geometric graphs. In *Proc. ICLR*, 2021. 2

[8] M. Defferrard, X. Bresson, and P. Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Proc. NeurIPS*, 2016. 2

[9] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Proc. CVPR*, 2009. 1, 5

[10] P. Ding, D. Soselia, T. Armstrong, J. Su, and F. Huang. Reviving shift equivariance in vision transformers. *arXiv preprint arXiv:2306.07470*, 2023. 2

[11] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proc. ICLR*, 2021. 1, 2

[12] H. Fan, B. Xiong, K. Mangalam, Y. Li, Z. Yan, J. Malik, and C. Feichtenhofer. Multiscale vision transformers. In *Proc. CVPR*, 2021. 1, 2, 18

[13] J. Hartford, D. Graham, K. Leyton-Brown, and S. Ravanbakhsh. Deep models of interactions across sets. In *Proc. ICML*, 2018. 2

[14] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proc. CVPR*, 2016. 2

[15] C. Kanbak, S.-M. Moosavi-Dezfooli, and P. Frossard. Geometric robustness of deep networks: analysis and improvement. In *Proc. CVPR*, 2018. 21

[16] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. In *Proc. ICLR*, 2017. 2

[17] D. M. Klee, O. Biza, R. Platt, and R. Walters. Image to sphere: Learning equivariant features for efficient pose prediction. In *Proc. ICLR*, 2023. 2

[18] A. Krizhevsky, G. Hinton, et al. Learning multiple layers of features from tiny images. 2009. 1, 5

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *Proc. NeurIPS*, 2012. 2

[20] Y. Li, C.-Y. Wu, H. Fan, K. Mangalam, B. Xiong, J. Malik, and C. Feichtenhofer. MViTv2: Improved multiscale vision transformers for classification and detection. In *Proc. CVPR*, 2022. 1, 2, 5

[21] M. Lin, Q. Chen, and S. Yan. Network in network. *arXiv preprint arXiv:1312.4400*, 2013. 2

[22] I.-J. Liu, R. A. Yeh, and A. G. Schwing. PIC: permutation invariant critic for multi-agent deep reinforcement learning. In *Proc. CORL*, 2020. 2

[23] I.-J. Liu, Z. Ren, R. A. Yeh, and A. G. Schwing. Semantic tracklets: An object-centric representation for visual multi-agent reinforcement learning. In *Proc. IROS*, 2021. 2

[24] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proc. ICCV*, 2021. 1, 2, 3, 5, 7, 18

[25] Z. Liu, H. Hu, Y. Lin, Z. Yao, Z. Xie, Y. Wei, J. Ning, Y. Cao, Z. Zhang, L. Dong, et al. Swin transformer v2: Scaling up capacity and resolution. In *Proc. CVPR*, 2022. 1, 2, 3, 5, 18

[26] H. Maron, H. Ben-Hamu, N. Shamir, and Y. Lipman. Invariant and equivariant graph networks. In *Proc. ICLR*, 2019. 2

[27] H. Maron, O. Litany, G. Chechik, and E. Fetaya. On learning sets of symmetric elements. In *Proc. ICML*, 2020. 2

[28] H. Michaeli, T. Michaeli, and D. Soudry. Alias-free convnets: Fractional shift invariance via polynomial activations. In *Proc. CVPR*, 2023. 2

[29] C. Morris, G. Rattan, S. Kiefer, and S. Ravanbakhsh. SpeqNets: Sparsity-aware permutation-equivariant graph networks. In K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, editors, *Proc. ICML*, 2022. 2

[30] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. PointNet: Deep learning on point sets for 3D classification and segmentation. In *Proc. CVPR*, 2017. 2

[31] M. A. Rahman and R. A. Yeh. Truly scale-equivariant deep nets with Fourier layers. 2024. 2

[32] S. Ravanbakhsh, J. Schneider, and B. Póczos. Equivariance through parameter-sharing. In *Proc. ICML*, 2017. 2

[33] S. Ravanbakhsh, J. Schneider, and B. Poczos. Deep learning with sets and point clouds. In *Proc. ICLR workshop*, 2017. 2

[34] R. A. Rojas-Gomez, T. Y. Lim, A. G. Schwing, M. N. Do, and R. A. Yeh. Learnable polyphase sampling for shift invariant and equivariant convolutional networks. In *Proc. NeurIPS*, 2022. 2, 5, 19

[35] R. A. Rojas-Gomez, R. A. Yeh, M. N. Do, and A. Nguyen. Inverting adversarially robust networks for image synthesis. In *Proc. ACCV*, 2022. 21

[36] D. Romero, E. Bekkers, J. Tomczak, and M. Hoogendoorn. Attentive group equivariant convolutional networks. In *Proc. ICML*, 2020. 2

[37] D. W. Romero and S. Lohit. Learning partial equivariances from data. In *Proc. NeurIPS*, 2022. 2

[38] H. Salman, A. Ilyas, L. Engstrom, A. Kapoor, and A. Madry. Do adversarially robust ImageNet models transfer better? In *Proc. NeurIPS*, 2020. 21

[39] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen. MobileNetV2: Inverted residuals and linear bottlenecks. In *Proc. CVPR*, 2018. 2

[40] M. Shakerinava and S. Ravanbakhsh. Equivariant networks for pixelized spheres. In *Proc. ICML*, 2021. 2

[41] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE SPM*, 2013. 2

[42] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proc. ICLR*, 2015. 2

[43] T. van der Ouderaa, D. W. Romero, and M. van der Wilk. Relaxing equivariance constraints with non-stationary continuous filters. In *Proc. NeurIPS*, 2022. 2

[44] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Proc. NeurIPS*, 2017. 1, 2

[45] S. R. Venkataraman, S. Balasubramanian, and R. R. Sarma. Building deep equivariant capsule networks. In *Proc. ICLR*, 2020. 2

[46] M. Vetterli, J. Kovačević, and V. K. Goyal. *Foundations of signal processing*. Cambridge University Press, 2014. 2

[47] M. Weiler and G. Cesa. General E(2)-equivariant steerable CNNs. In *Proc. NeurIPS*, 2019. 2

[48] H. Wu, B. Xiao, N. Codella, M. Liu, X. Dai, L. Yuan, and L. Zhang. CvT: Introducing convolutions to vision transformers. In *Proc. CVPR*, 2021. 1, 2, 5

[49] T. Xiao, Y. Liu, B. Zhou, Y. Jiang, and J. Sun. Unified perceptual parsing for scene understanding. In *Proc. ECCV*, 2018. 5, 7

[50] R. A. Yeh, Y.-T. Hu, and A. Schwing. Chirality nets for human pose regression. In *Proc. NeurIPS*, 2019. 2

[51] R. A. Yeh, A. G. Schwing, J. Huang, and K. Murphy. Diverse generation for multi-agent sports games. In *Proc. CVPR*, 2019. 2

[52] R. A. Yeh, Y.-T. Hu, M. Hasegawa-Johnson, and A. Schwing. Equivariance discovery by learned parameter-sharing. In *Proc. AISTATS*, 2022. 2

[53] M. Yi, L. Hou, J. Sun, L. Shang, X. Jiang, Q. Liu, and Z. Ma. Improved OOD generalization via adversarial training and pretraing. In *Proc. ICML*, 2021. 21

[54] M. Zaheer, S. Kottur, S. Ravanbakhsh, B. Poczos, R. R. Salakhutdinov, and A. J. Smola. Deep sets. In *Proc. NeurIPS*, 2017. 2

[55] R. Zhang. Making convolutional networks shift-invariant again. In *Proc. ICML*, 2019. 2, 5, 6, 15

[56] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba. Scene parsing through ADE20K dataset. In *Proc. CVPR*, 2017. 1, 7

[57] X. Zou, F. Xiao, Z. Yu, and Y. J. Lee. Delving deeper into anti-aliasing in ConvNets. In *Proc. BMVC*, 2020. 2